

A PROCESS FOR CREATING CELTIC KNOT WORK

A Thesis

by

HUNTER GUYMIN PARKS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

May 2003

Major Subject: Visualization Sciences

A PROCESS FOR CREATING CELTIC KNOT WORK

A Thesis

by

HUNTER GUYMIN PARKS

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Karen Hillier
(Chair of Committee)

Donald House
(Member)

Wayne Smith
(Member)

Phillip Tabb
(Head of Department)

May 2003

Major Subject: Visualization Sciences

ABSTRACT

A Process for Creating Celtic Knot Work. (May 2003)

Hunter Guymin Parks, B.F.A., Stephen F. Austin State University

Chair of Advisory Committee: Prof. Karen Hillier

Celtic art contains mysterious and fascinating aesthetic elements including complex knot work motifs. The problem is that creating and exploring these motifs require substantial human effort. One solution to this problem is to create a process that collaboratively uses interactive and procedural methods within a computer graphic environment. Spline models of Celtic knot work can be interactively modeled and used as input into procedural shaders. Procedural shaders are computer programs that describe surface, light, and volumetric appearances to a renderer. The control points of spline models can be used to drive shading procedures such as the coloring and displacement of surface meshes. The result of this thesis provides both an automated and interactive process that is capable of producing complex interlaced structures such as Celtic knot work within a three-dimensional environment.

TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES.....	v
CHAPTER	
I INTRODUCTION.....	1
II BACKGROUND.....	3
Contemporary Works Influenced by the Celtic Culture.....	16
III THE PROCESS.....	28
Maya.....	30
Renderman.....	32
IV RESULTS AND DISCUSSION.....	38
Critique of Visual Output.....	45
Usage of the Celtic Knot Work Process by Other Artists.....	46
V CONCLUSIONS.....	47
Future Work.....	48
REFERENCES.....	50
VITA.....	52

LIST OF FIGURES

FIGURE	Page
1 Ornamental Silver Disc First Half 1 st Century B.C.....	2
2 Courtney Davis <i>The Painted Warrior</i> 1996.....	6
3 Silver Torc Late 3 rd -early 2 nd Century B.C.....	8
4 Gold Torc End 6 th -early 5 th Century B.C.....	9
5 <i>Battersea Shield</i> 1 st Century B.C.....	9
6 Bronze Mirror End 1 st Century B.C. to Early 1 st Century A.D.....	10
7 Page from the <i>Book of Kells</i> Late 8 th to Early 9 th Century A.D.....	12
8 Carpet Page from <i>The Book of Durrow</i> About 675 A.D.....	13
9 Eight Elementary Knots.....	15
10 Steven O’Loughlin <i>LA Riots</i> 1992.....	17
11 Jen Delyth <i>Gavrinis</i> 1995.....	18
12 A. Andrew Gonzalez <i>The Summoning of the Muse</i> 1993.....	19
13 Brian Froud <i>Twilight</i> 1976.....	20
14 Output from <i>Knot Assistant</i>	22
15 Mathematical Knots Output from <i>KnotPlot</i>	24
16 Output from Aaron Otstott’s Procedural Celtic Shader.....	25
17 Output from the Ornamentation Process.....	27
18 Spline Curves and their Control Points.....	29
19 Control Panel and Curves Formed within Template Area.....	31

FIGURE		Page
20	Finding the Shortest Distance between the Spline and the Current Shading Position.....	36
21	Celtic Knot Work Shader 01.....	40
22	Celtic Knot Work Shader 02.....	40
23	Celtic Knot Work Shader 03.....	41
24	Celtic Knot Work Shader 04.....	41
25	Shader with Segmentation.....	42
26	Celtic Shader with Threading.....	42
27	Computer Animation without Celtic Shaders.....	43
28	Computer Animation with Celtic Shaders.....	43
29	Shading Error 01.....	44
30	Shading Error 02.....	44

CHAPTER I

INTRODUCTION

Since the beginning of recorded history, there have existed many cultures, each having a unique presence. Artists and art forms of the present are influenced and inspired by exposure to these cultures. One such culture was that of the Celts, which produced mysterious works of art containing complex interlacing designs. The Celtic culture was among European Iron Age peoples who shaped metal, stone, wood, and other natural materials into sculpture, weaponry, and ornamentation (Figure 1). These works represent a transitional period between primitive art of prehistoric times and advanced art forms of the Renaissance. During this transitional period the technology and skills of Celtic peoples were evolving. This is reflected in the execution of many works containing imperfections which give them a loose and painterly appearance. These artifacts also held a truth to materials, meaning the artist worked with the material in order to create something new but also preserved the natural essence of the medium.

The subject matter of Celtic Art usually involves mythical themes which was a result of their unique state of mind. The Celts felt a supernatural presence within the natural forms that surrounded them. These beliefs were echoed in creations referencing the forests and creatures that dwelled within them. The combination of such beliefs and methods of execution seem to capture a sense of mysticism within Celtic work. It is a

This thesis follows the style and format of *ACM SIGGRAPH*.

sense of mysticism that this thesis project attempts to capture within its visual output.

The cultural myths and aesthetic elements of Celtic Art have inspired this thesis project to create Celtic forms in Computer Graphics. The difficulty in exploring Celtic designs is that they contain complicated interlacing forms that require substantial human effort to create. One method of exploration is to create a digital process that collaboratively uses interactive and procedural methods. Interactive methods give artists the freedom to create desired forms, while procedural methods can be used to automate the production of complex and repetitive detail. The goal is to give artists high level control and relief from tedious modeling tasks.



Figure 1: Ornamental Silver Disc First Half 1st Century B.C.

CHAPTER II

BACKGROUND

The term Celtic Art is generally applied to both the art produced in Europe between the fifth century B.C. and the first century A.D. by Iron Age peoples commonly labeled Celts and the art created in Britain and Ireland between the fifth and the twelfth centuries A.D. outside of the areas of Anglo-Saxon settlement. The word Celt is derived from the Greek name for barbarians living in Europe, and the cultural concept of Celts is defined as speakers of a group of related languages. Although there is no such a thing as a Celtic race or Empire, there existed a Celtic culture that began in prehistoric times and ended in post-Medieval Age. Reminders of the Celtic culture can still be seen across Europe in the names of towns, rivers, and mountains: *London, Dublin, Milan, and Paris* [14].

The origin of the Celtic Culture is dated around the fourth century B.C. The Celts were Pagan until introduced to Christianity in the fourth and fifth centuries A.D. The Pagan Celts included two casts of people, the tribes and the druids. The druids were the unifying force between different Celtic tribes. The origin of the druids and their religion likely came from the megalithic priesthood. Megalithic priests were those who used large stone monuments such as Stonehenge within their rituals, and were most likely in existence prior to 2000 B.C. Around 600 B.C. the druids abandoned the great stones and reverted to natural shrines such as trees. Their name is thought to have come from “Drus”, the ancient name for an oak tree that is sacred to them. Druids were the

teachers, seers, poets, judges, and doctors of the Celtic culture and were recruited among the ranks of nobility. Their education was long and involved decades of memorizing sacred texts which religious taboo prevented from being in written form. In their religious roles, the druids insured the conduct of religious practices and presided over sacrificial rites. The specifics of their religious activities remain shrouded in mystery because information was transmitted orally. What is generally known about their religion, taken from classical writings, is that they held a close attunement to nature, the seasons, and all aspects of the natural world. They associated natural features like springs, rivers, and mountains with super-natural deities who controlled them. They also believed that the life of the soul was not interrupted by death but would continue in two ways, either dwelling in a spiritual realm or returning to earth in the reincarnation of the soul. This belief was reinforced by the cycle of the seasons. Every year the trees shed their leaves, vegetation withered, and once winter came there was little sign of life from nature. Yet with the coming of spring rebirth occurred, and the forest came to life again. Naturally, they believed that this cycle applied to humans as well [7, 13].

The other division of Pagan Celts was the tribes. The tribes consisted of nomadic groups that were spread across Europe. At almost any period, these tribes were involved in war with other Celtic tribes or with large empires such as the Romans. Posidonius, a classical writer, observed the Celts during his travels and describes them as having a courageous and receptive temperament. They had a passion for ornaments and a constant obsession with honor. He also observed that they had a custom of preserving the heads of their enemies killed in battles as trophies. These heads were one of the

most valuable possessions a Celt could own because it was proof of honor and courage. Accounts of the early Celts were also documented by Julius Caesar in the Gallic Wars. He reported that they were a wild and war-faring people who displayed an overwhelming trait of flamboyance. Both men and women fought in wars. Before going into battle they would form their hair into a spiked mane and paint elaborate designs on their bodies with bluish-gray dye (Figure 2). This was the psychology of Celtic warfare. They wanted to project a larger than life appearance in order to intimidate their opponents [5, 13].



Figure 2: Courtney Davis *The Painted Warrior* 1996

The artwork crafted by the Pagan Celts consisted of jewelry, sculpture, weaponry, and primitive scientific instruments used by the druids. Their style of art was influenced by classical Germanic, Greek, Oriental, Roman, and Viking art. Most surviving work is in the form of bronze; but they also used gold, silver, iron, stone, and wood. The surfaces of Pagan Celtic artifacts contain curvilinear patterns that are often found in arrangements of three. One reoccurring motif is tightly coiled spirals appearing in triskel formations. The purpose of these designs is not precisely known; but it is thought that depicting natural forms was taboo [13]. This would explain why most Celtic art of the Pagan era contains abstract imagery. An example of early Celtic art is the torc (Figures 3 and 4) made throughout Europe in the latter part of the first millennium B.C. The torc is a circular band often made from several separate interlaced strands of metal. Torcs did not join to form a complete circle. Instead, the two ends, called finials, were left open and lavishly decorated. Because they were left open, the torc could be adjusted to fit around different parts of the body like the arm, leg, or neck. Torcs were a symbol of wealth and social status, and would often be worn by a Celtic chief and those in positions of power. The Roman historian Dio Cassius describes Boudicca, queen of the Iceni Celts, as riding into battle wearing a great twisted golden torc around her neck [5].

Bronze shields (Figure 5) have also been found that were crafted in the early Celtic era. The Battersea shield is a Celtic shield dredged from the Thames River in 1851, and its origin is dated around the 1st century B.C. The shield is made of sheets of bronze bound and riveted to wood or leather backing. The shield is composed of three

roundels, a circular area containing decorative content, and red glass inlay that was cast and riveted to the main surface. The three roundels contained within the shield is an example of the symbolism of three-ness that is found in many early Celtic motifs [16]. Bronze mirrors, (Figure 6), which have been found in Britain and Europe, were another popular craft of the early Celts. They were made until the first century A.D. and consisted of a circular disc of bronze with an ornamental handle. One side is highly polished and lacks decoration, while the reverse side contains elaborate design of interlaced curves and spirals along with fine detailed crosshatching in negative areas [5]. Large collections of Celtic Art from the Pagan era can be found in the British museum, London.



Figure 3: Silver Torc Late 3rd-early 2nd Century B.C.



Figure 4: Gold Torc End 6th-early 5th Century B.C.



Figure 5: *Battersea Shield* 1st Century B.C.



Figure 6: Bronze Mirror End 1st Century B.C. to Early 1st Century A.D.

The next phase of evolution in Celtic culture was the gradual conversion to Christianity that was introduced to them in the fourth and fifth centuries A.D. The conversions of Celts to Christianity marked the beginning of a new style of Celtic art influenced by Germanic styles, Mediterranean styles, and Christian subject matter. Christianity was established in Roman Britain in the fourth century A.D. In the second half of the sixth century, monasticism, which had originated in the Mediterranean, spread through the Celtic areas. As a result, monks produced most Celtic art of that period. These monks created pictorial gospels called illuminated manuscripts that contained

elaborate ornamentation and gilded illustrations. It is in Illuminated Manuscripts that complex interlaced knot work began to appear.

The Book of Kells is an illuminated manuscript created in the late eighth, early ninth century A.D, and is regarded as one of the finest achievements of Celtic art. *The Book of Kells* was created by Columban monks who lived on the remote island of Iona, off the west coast of Scotland. Columban monks were disciples of Saint Columba, who founded many monasteries in Celtic regions and promoted the practice of illumination. *The Book of Kells* displays a fusion of Celtic, Germanic, and Mediterranean styles in the form of chord interlace, animal interlace, and rectilinear patterns. The page shown in (Figure 7) displays various types of interlaced knot work contained within the border of the page. Elsewhere there are tiny heads, busts and animals appearing within the assortment of motifs [16].

The Book of Durrow, created around 675 A.D., is another illuminated manuscript displaying elaborate ornamentation. It is not precisely known, but this manuscript was supposedly created by the Columban monks of Iona. The decoration is painted in red, yellow, green, and deep browns. Ornamentation within the *Book of Durrow* displays trefoil motifs, ribbon interlace, animal interlace, and rectilinear patterns. Examples of early Celtic influence appear in pages containing coiled spirals forming triskel structures (Figure 8). This motif reflects designs found in Iron Age artifacts and shows the influence of Pagan Celtic tradition among illuminated manuscripts [16]. Collections of illuminated manuscripts can be found in the Royal Irish Academy, Dublin, and the University of Oxford, Oxford.

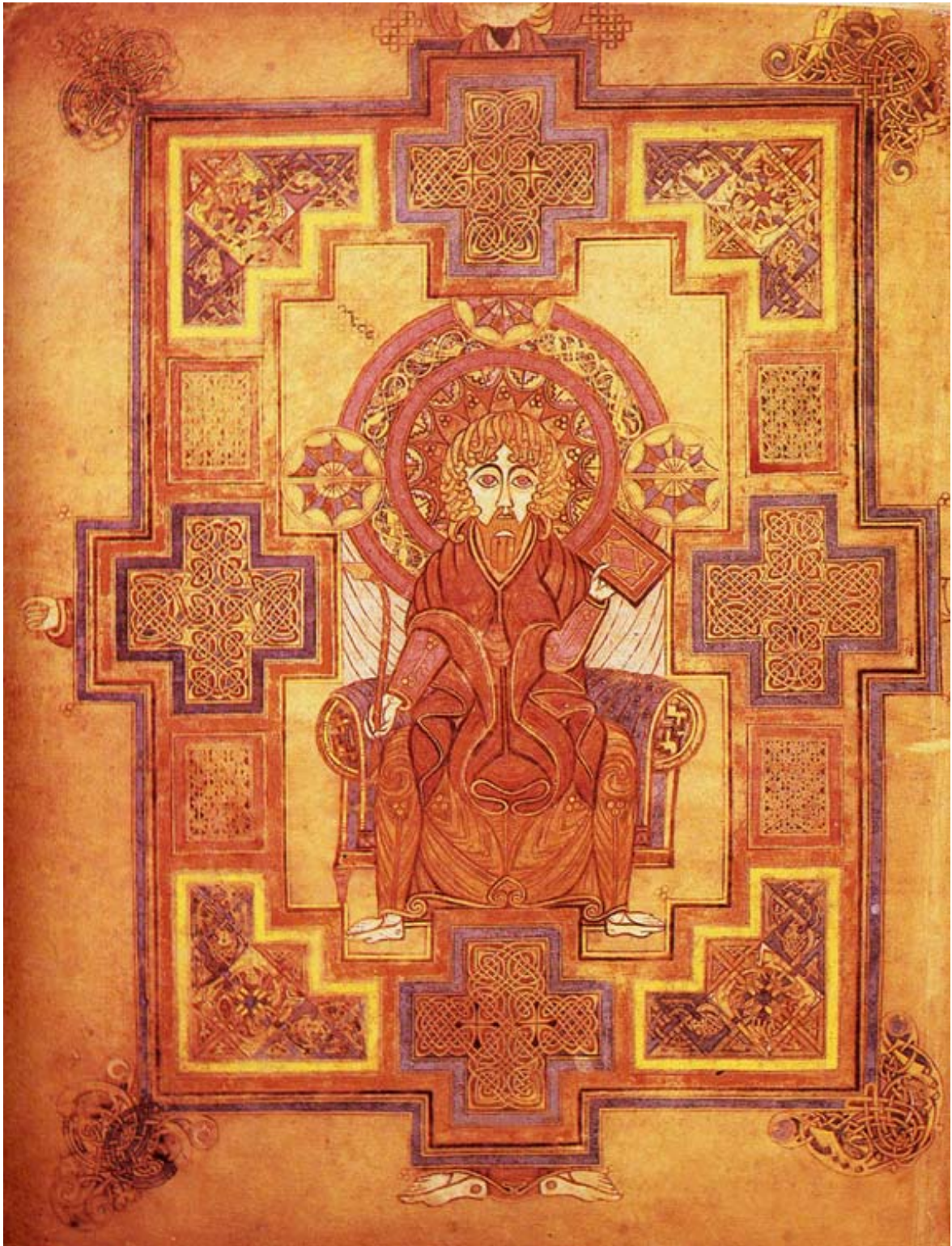


Figure 7: Page from the *Book of Kells* Late 8th to Early 9th Century A.D.



Figure 8: Carpet Page from *The Book of Durrow* About 675 A.D.

The Celtic motif in Pagan and Christian times has a geometric appearance in the form of interlacing knot-work, spirals, or interlacing human, bird, and beast figures. Knot-work is the term used to describe the appearance of loose knots caused by breaking and rejoining some of the cords into a plait. Knot-work evolved from the plait, which can be described as a single cell cutout from a grid. Chords within a cell are cut and rejoined by short curves creating a continuous path. Eventually, the Celts discovered how to devise new patterns from basic plaits by making breaks in the design leading to more complicated forms of interlaced ornamentation [3]. Eight elementary patterns (Figure 9), developed from plaits, have been identified which form the basis of nearly all interlaced patterns in Celtic knot work [1]. There exist many different designs created from breaking and rejoining chords, but exploring them involves tedious work. However, all structures contain reoccurring patterns that can be isolated and used to reconstruct the network that they describe. Fast, automated procedures can be developed to handle such repetitive tasks and allow artists to explore Celtic designs within a computer graphic environment.

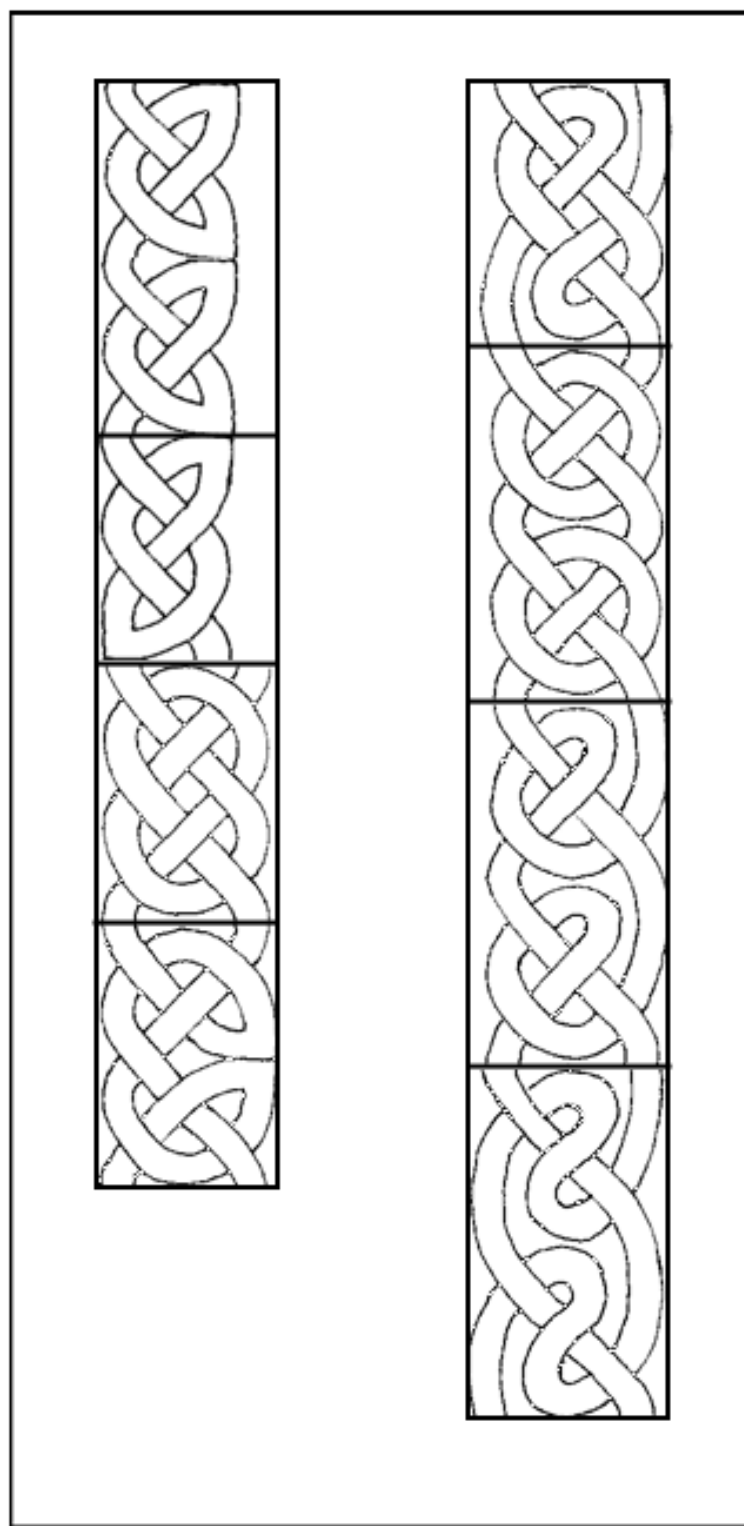


Figure 9: Eight Elementary Knots

Contemporary Works Influenced by the Celtic Culture

Steven O'Loughlin is a contemporary artist whose work is influenced by a mixture of urban and cultural elements including Celtic Art. Celtic art being one of the dominate influences, his paintings also include Mexican, Asian, and African styles. The influence of Celtic Art in O'Loughlin's work comes from his Irish heritage and a fascination with interlacing forms. This can be seen in *L.A. Riots* (Figure 10) where figures are arranged in a bizarre tangle of interlaced human limbs. The influence of Hispanic themes within his paintings comes from being exposed to Latin cultures within his home of Los Angeles. This can be seen in works containing bright color palettes and Hispanic content. The subject matter contained within his paintings often conveys narratives dealing with the social and aesthetic issues of his surrounding environment. Some of these issues include the Los Angeles riots of 1992 and the California fires and earthquakes of the early 1990's. Overall, O'Loughlin's work exhibits flamboyant and graphic designs that display an interesting mix of ethnic influences [17].

Jen Delyth is another contemporary artist whose work is influenced by Celtic Art (Figure 11). Her work is inspired by the folk motifs of ancient Celtic art and the artist's experience as a woman in the 21st century. Delyth's interest in Celtic mythology comes from the folk traditions of her Welsh background. Her paintings incorporate ancient curvilinear motifs, conveying archetypal, spiritual, and mythical themes. Delyth also uses a cool color palette containing subtle areas of warmth. These earthy colors reflect the veil of moss, peat, and grass that covers her surrounding environment of Southern Wales [6].



Figure 10: Steven O'Loughlin *LA Riots* 1992



Figure 11: Jen Delyth *Gavrinis* 1995

Celtic themes also have a strong presence within Fantasy art. Fantasy art is a broad style of art that contains mystical and surreal elements. The influence of the Celtic culture is directly seen in fantasy works that contain aesthetic and mythological

Celtic references. One such fantasy artist whose work exhibits a Celtic influence is Brian Froud (Figure 13). Froud is most recognized for his concept work on the movies *The Labyrinth* and *The Dark Crystal*. His works involve humans and creatures that dwell within mystical realms. Deities of Celtic mythology inspired some of these creatures. Knot work motifs are also incorporated into the clothing and armor of his characters [8]. Celtic art has also had an indirect influence upon artists such as A. Andrew Gonzalez. Gonzalez's work is inspired by mystic themes and earlier Fantasy artists including H.R. Giger. Gonzalez's paintings contain interlocking human figures surrounded by interlaced forms that coil into spiraling structures [10]. The artist's compositions reflect the interlaced human and animal motifs found in medieval illuminated manuscripts (Figure 12).



Figure 12: A. Andrew Gonzalez *The Summoning of the Muse* 1993



Figure 13: Brian Froud *Twilight* 1976

Artist, computer scientist, and writer Andrew Glassner created a computer program called *Knot Assistant* that lets users interactively create two-dimensional Celtic knot work (Figure 14). Glassner's work is based on the work of George and Ian Bain who devised a simple method of constructing Celtic knot work by hand. Their methods are based on using grid cells, reference skeletons, and plait breaking rules to create knot compositions. Glassner observed that the mechanics of this process could fit into the framework of a computer program. A user arranges breaks within a construction grid, and a reference skeleton is created procedurally according to the placement of the breaks. The reference skeleton is then used to create Bezier curve segments along the path of the skeleton. Overlapping is controlled by the location of grid cells, even or odd, and directional procedures that track the direction that the chord is traveling. Within the program a user can control the size of the primary grid, draw and modify break lines, set the size of overlapping gaps, and control the thickness of the rendered chord. *Knot Assistant* outputs rendered chords as a postscript file that can be imported into other graphics programs for further enhancement. Glassner also experimented with ways of translating output from his program into three dimensions. He wrote an additional procedure which creates a Bezier curve that travels through the centerline of the chord. This spline can then be imported into a three-dimensional software package and used as a path in order to create lofted surfaces [9].

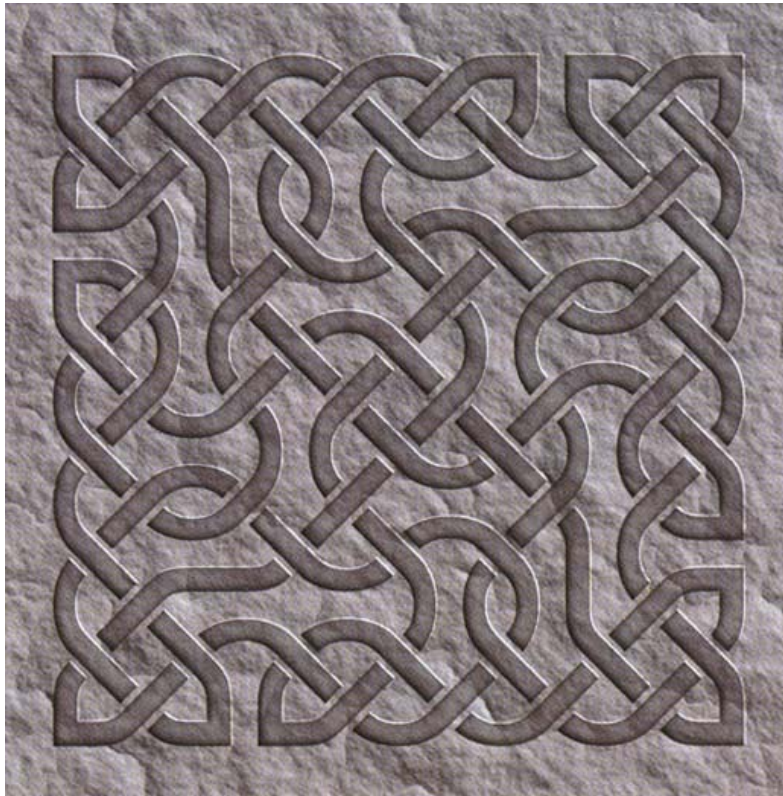


Figure 14: Output from *Knot Assistant*

Rob Scharein of Imager Computer Graphics Laboratory, Department of Computer Science, University of British Columbia, is the author of *Knot Plot*. *Knot Plot* is a program for interactively visualizing and manipulating mathematical knots in three and four dimensions (Figure 15). The *Knot Plot* features consist of a large database of tame knots and links that can be loaded into the program. The knots are classified in order of increasing complexity. One measure of complexity is the crossing number, or the number of double points in a planar projection of the knot. Once the knots are loaded, new knots can be created from old knots by using *Knot Plots* transformation tools and changing knot calculation attributes.

Knot Plot is based on Rob Scharein's research into knot theory. Knot theory is a branch of algebraic topology that deals with the embedding of one topological space into another. A knot is defined to be a closed piecewise linear curve in three-dimensional Euclidean space. Knots studied in knot theory almost always have connected ends, creating a closed loop. So the common idea of a knot as a tangled cord with two loose ends is different from the mathematical definition of one. Mathematical knots in knot theory are classified as either tame or wild. A tame knot is one that can be represented as a polygonal path in three-dimensional space. In other words, tame knots are knots that can be constructed from a finite length. Those that cannot be classified as tame are called wild knots. Wild knots are those that cannot be represented as a polygonal path in three-dimensional space, and might contain an infinite number of links or crossing loops. *Knot Plot* is only capable of visualizing tame knots [20].

Knot Plot can create a variety of three-dimensional knot arrangements by using mathematical descriptions. Scharein experimented with creating Celtic knot work by splicing together small knots under a symmetry group. He noted, though, that *Knot Plot* was not designed to fully describe Celtic knot work. Its main purpose is to create mathematical knots which are visually interesting but difficult to manipulate into desired forms.

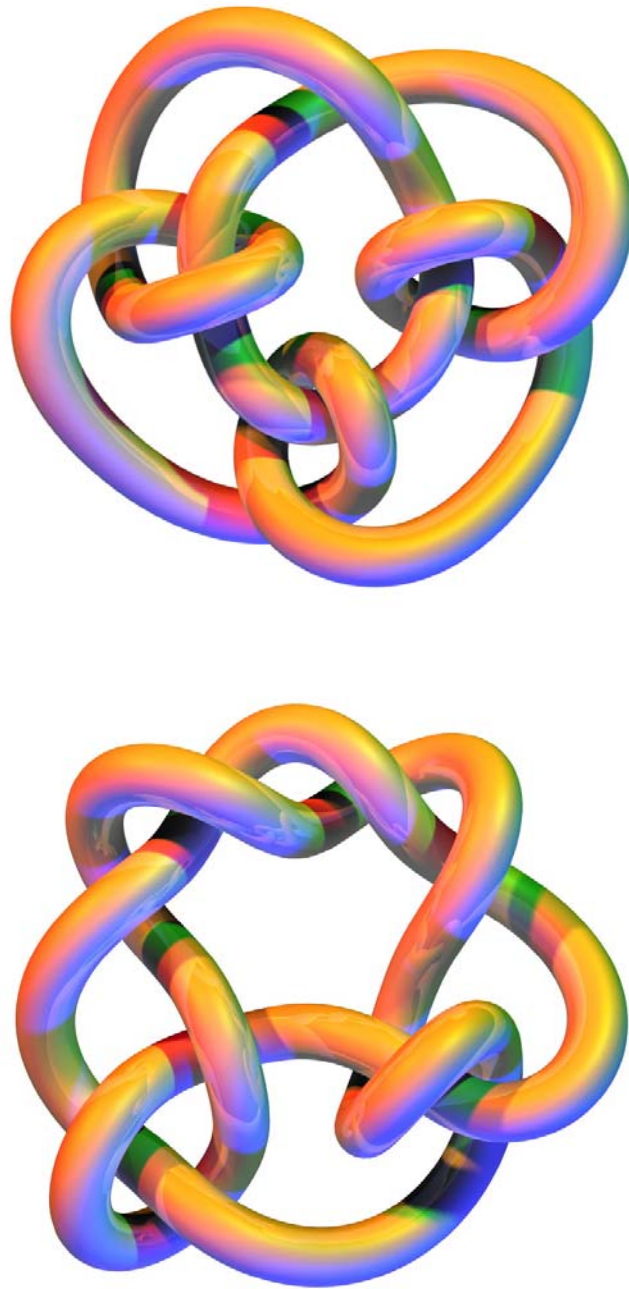


Figure 15: Mathematical Knots Output from *KnotPlot*

Aaron Otstott, a graduate student in the Visualization Program at Texas A&M University, developed a procedural Renderman shader that produces a tile-able Celtic braid pattern (Figure 16). Renderman is a standard communications protocol between modeling and rendering programs that is capable of producing photorealistic images. Renderman contains a shading language that is used to describe surface, light, and volumetric appearances to a renderer. User-defined parameters within Otstott's shader include the ability to control line-width, frequency, and the conversion of Cartesian to polar coordinates. The shader uses an array of control points that acts as a skeleton for shading procedures to follow. The control points are organized in layers so that when evaluated, overlapping lines are produced. Each layer is also assigned a specific color that is applied to the rendered line stroke. The other procedure of this shader is its ability to convert Cartesian into polar coordinates. When activated, the shader references a polar coordinate system resulting in pattern construction around an arbitrary axis. The visual output of the polar coordinate procedure produces a radial braid instead of a linear pattern [18].



Figure 16: Output from Aaron Otstott's Procedural Celtic Shader

Wong, Zongker, and Salesin, from the University of Washington, developed a process that creates two-dimensional ornamentation algorithmically (Figure 17). They introduce the idea of adaptive clip art, which attempts to capture the essence of ornamentation, encoding it as a set of rules for creating specific ornamental patterns. The process created can generate ornamentation within a specified region. If the region is reshaped, the ornamentation will regenerate to conform to its new boundaries.

Their process uses Perl as a preprocessor, and C++ for the frame work. The preprocessor reads a rule file, encodes an ornamental pattern, and outputs a source file. When the source file is compiled, a region in which ornamentation is to appear can be input into the executable. A reference skeleton is then created and composed as a set of two-dimensional curves and filled boundaries. Growth of the skeleton begins in large open spaces and then shifts to filling in corners of the region. Ornamental clip art is then arranged along the path of the skeleton according to principles of traditional ornamentation. These principles include translating, rotating, scaling, mirroring, or inversing clip art along the skeletons path.

Wong, Zongker, and Salesin's process outputs a postscript file containing the outline of each ornamental element. Most imagery produced displays hierarchical growth where patterns start from seed points and branch off into finer detail. The user can attach arbitrary C++ code to customize the appearance of the ornamentation or import the file into other graphics programs for further enhancement [19].

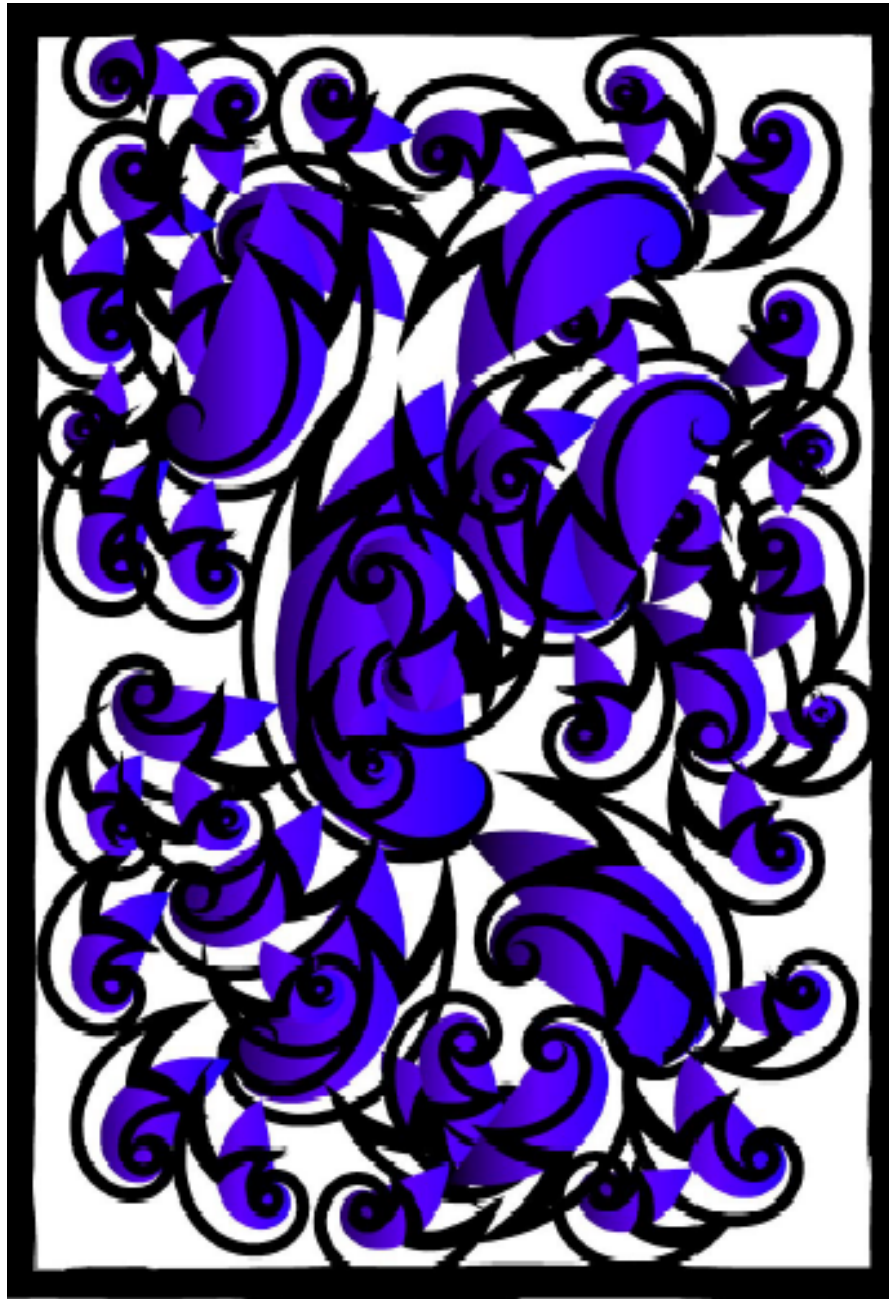


Figure17: Output from the Ornamentation Process

CHAPTER III

THE PROCESS

The process developed for this thesis project is composed of interactive and procedural methods that work collaboratively in order to build complex forms inspired by Celtic knot work. The process begins with the modeling of Celtic knots within an interactive environment and then proceeds with the processing of modeling data within shader procedures. Spline models created by the user are used to describe Celtic knots.

A spline is a curve described by an equation, generally falling into two broad categories: interpolating and approximating. Interpolating curves (Figure 18(a)) will pass through the control points used to define them, while an approximating curve (Figure 18(b)) will only get near to the points. The term spline originated in manual drafting, where a spline is a thin strip of material. This strip is held in place by weights in order to create a curve that can be traced [4]. Antonio Gaudi used physical splines to visualize the unusual vaulted ceilings of the cathedral *La Sagrada Familia*, 1884-present. He would suspend strands of twine and hang weights on them. This would create upside down vaults that he could experiment with and reference while designing the cathedral.

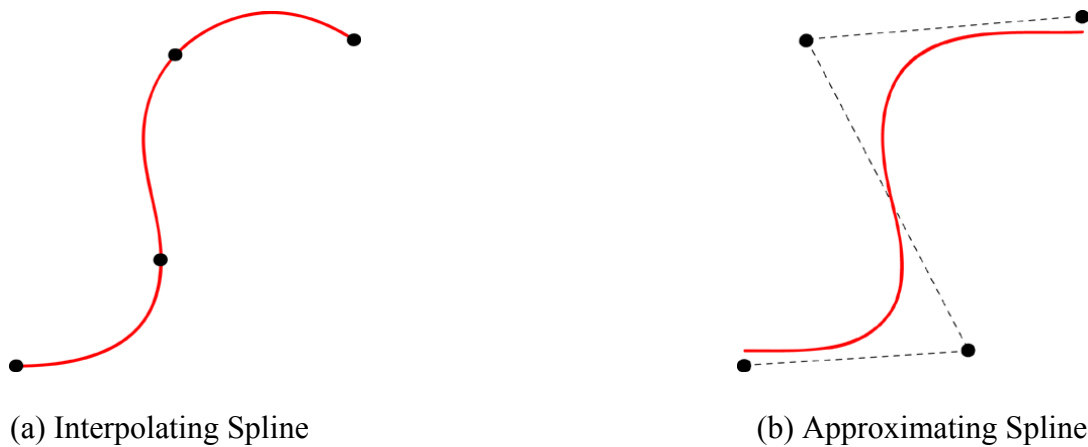


Figure 18: Spline Curves and their Control Points

The development of curves for use in computer graphics generally began with the early work of S.A. Coons and P.E. Bezier. They introduced the use of nonlinear, parametric, polynomial representations for segments from which piecewise curves are assembled. Bezier's work also introduced the techniques for approximating specified points called control points. Two types of splines that evolved from curve research are B-splines and Catmull-Rom splines. B-splines, introduced to computer graphics by R.F. Riesenfeld, are a type of local approximating parametric curve. The Catmull-Rom spline, introduced by E. Catmull and R. Rom, is a type of local interpolation spline developed from the B-spline [4, 12].

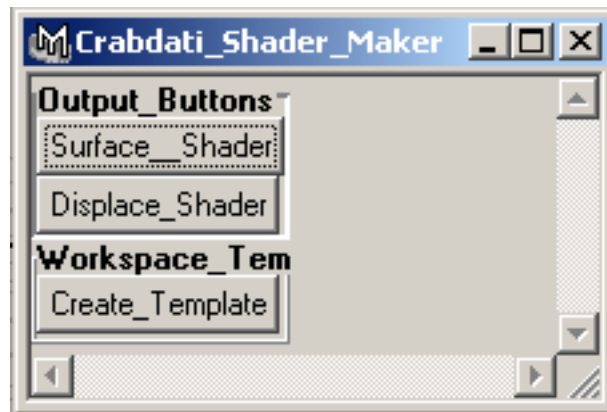
Splines began to be used for computer graphics purposes in the late 1960's by engineers in the French auto industry to describe contours of automobiles [4]. Now, various types of splines are used to describe curves and surfaces within computer-aided modeling programs. Maya, Alias, and SoftImage are examples of sophisticated

modelers that allow for the construction of complex geometry using spline representations.

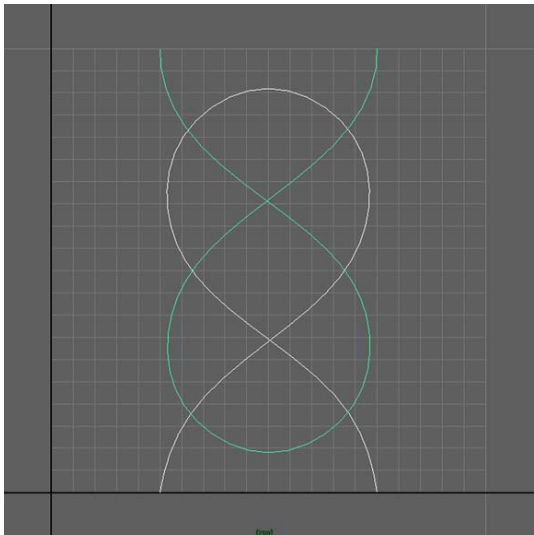
Maya

For the purpose of interactively modeling splines, Alias/Wavefront's Maya was chosen to be the modeler for this thesis project. Maya contains sophisticated modeling tools that provide a wide-range of interactivity and freedom for creating three-dimensional geometry. Its spline creation and editing tools allow the user to create multiple splines in three-dimensional space and to add, subtract, and translate control points along the spline. Maya also supports an embedded command and scripting language called MEL that allows direct control over Maya's features, processes, and workflows. The tools that Maya lacked for this thesis project were added using MEL scripts.

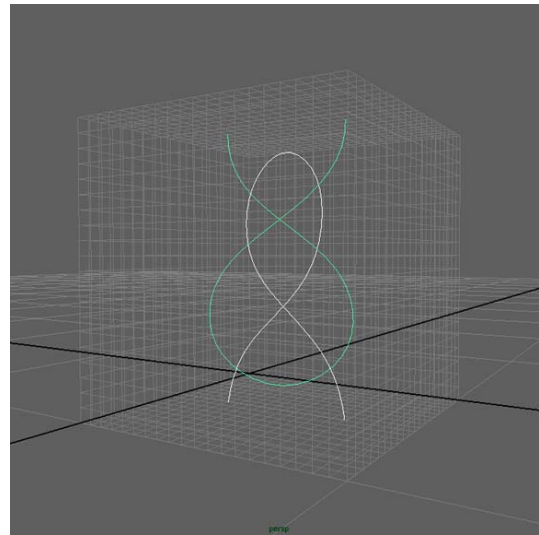
A custom control panel (Figure 19(a)) containing three controls was built for this process to aid in the construction and output of splines. The template control creates a $1.0 \times 1.0 \times 1.0$ polygonal cube within which the user can work. The purpose of this is to provide input for the shader that is within a normalized range, from 0 to 1.0, and to provide a construction reference when building the spline. Figure 19(b) and Figure 19(c) show two views of this template area in use, with a developing curve drawn inside. Splines can be drawn outside of the template area for unrestrained drawings; but in order to build complex models of Celtic knot work, a template area helps guide the user.



(a) Control Panel



(b) Orthographic View



(c) Perspective View

Figure 19: Control Panel and Curves Formed within Template Area

The surface and displacement output controls are used to create a new shader. Several MEL scripts are used to drive the output controls. The body of the shader, the shader procedures, are pre-written and stored in a text file. When the output controls are activated, file Input/Output commands within MEL are called to create a new text file and write the body of the shader to this file. Then, other MEL scripts are called to retrieve information about the modeled splines. MEL procedures march through the splines that have been selected and record the control points from which they are composed. These control points are then written to the new shader file along with the body of the shader. The shader procedures use these control points as input in order to perform shading calculations. The new shader can now be compiled and used by Renderman.

Renderman

The majority of this process is driven by procedural methods that were built using the Renderman shading language. Pixar's Renderman, is a standard communications protocol between modeling and rendering programs and is capable of producing photo-realistic quality images. Renderman is separated into two collaborative sections: scene modeling and appearance modeling. The placement and characteristics of objects, lights, and cameras in a scene are described to a renderer through the Renderman Interface. The details of the appearance for those objects and lights are described to a renderer by the Renderman Shading Language. The Renderman Interface is a scene description API, the set of data types and function calls that are used to

transmit data from one part of a system into another. RIB, Renderman Interface Bytestream, is the metafile format for the Renderman API. Metafiles are datafiles that encode a log of the calls to a procedural API. The Renderman Shading Language is a programming language used to describe the interaction of lights and surfaces. The programs containing these descriptions are called shaders. Shaders can describe the appearance of lights, surfaces, and volumes, which extends the types of materials and light sources available to a rendering system [2].

Two types of shaders are surface and displacement shaders. A surface shader describes the surface, color, and opacity in collaboration with incoming light properties. Displacement shaders calculate how much a point on the surface of geometry will shift from its original position. Shaders describe appearances procedurally or by referencing image files. A procedural shader uses an algorithmic approach, applying mathematical expressions in order to calculate information such as texture coordinates and pattern generation. The advantage of procedural shaders is that they can be used to describe meticulous patterns that would require repetitive and tedious effort to physically paint. Another advantage of procedural methods is that they preserve detail at all scales. Stored texture images have a fixed and limited resolution, so there is a lack of detail when they are stretched over large surfaces. The disadvantage of procedural shaders is the difficulty involved in writing a procedure that successfully describes a surface. They also frequently produce reconstruction artifacts. This is because procedural shaders sample the surface a finite number of times during rendering. If a procedural shader generates a high level of detail, some of this detail is lost between samples [2, 15].

A specialized Renderman shader was developed for this thesis project. Its algorithms were designed to evaluate control points extracted from the splines defined in Maya and manipulate them according to shader parameters. These parameters are defined internally but can be altered externally through the Renderman Interface. The parameters include `Km`, `direction`, `subd_level`, `freq`, `chord_width`, `offset`, `check_angle`, `thread_number`, `dispspace`, and `truedisp`. `Km` controls the magnitude level, `direction` specifies either horizontal or vertical mapping, and `freq` controls the number of times the design will be tiled. `Subd_level` controls how many times the spline will be subdivided, which determines how smooth the chord will appear, and `chord_width` controls the width of the line stroke region. The `offset` parameter is a dynamic parameter that allows the chord design to be animated. This parameter offsets subdivision calculations resulting in output that consistently varies over time. This is achieved by attaching an expression to the `offset` parameter that increments it each frame of an animation. The remaining parameters are `check_angle`, which switches on or off a procedure that checks the angle of splines to prevent rendering errors; `thread_number`, which controls how many threads will travel down the chord path; `dispspace`, which defines the coordinate space to be used in displacing calculations; and `truedisp`, which is a flag that tells the shader to either use bump or displacement mapping.

The control points listed in the shader describe the spline models that were created in Maya. They are inserted into a point array that contains three components per array element, the X, Y, and Z coordinates of the point. The control points are written

out in the order that they exist along the spline, and in the order that each spline or splines were selected. Since control points from multiple splines are stored in one array, flags are used to separate one spline's control points from another's. When the shader is output from Maya, the unique coordinate (1000.0, 1000.0, 1000.0) is inserted between each spline and used as a flag. When the control points are packed into the array they are ready to be sent into evaluation procedures.

The first operation within the procedure sets the direction and frequency of the output by manipulating the 2D texturing coordinates of the position being shaded. These texturing coordinates are global variables referred to as (s, t) within the Renderman shading language. The `offset` parameter is then evaluated and manipulates the subdivision algorithm if activated. The goal of the subdivision algorithm is to find the shortest distance between the current shading position (s, t) and the spline. This algorithm works by evaluating one spline segment at a time. First, four control points are inserted into a B-spline definition. A B-spline definition is used since Maya's curve tools use the same approximation technique. Other definitions could be used such as the Catmull-Rom basis, but this would interpolate the control points and create unpredictable output. The subdivision procedure is shown in Figure 20. It subdivides the spline segment into two regions, and tests are made to find which region is closest to the shading position (s, t) . This process is repeated `subd_level` number of times. The accuracy of the measurement depends on the number of subdivisions calculated. After the shortest distance is calculated, a shading method can be assigned to the shading position. If the shortest distance from the current shading position to the spline is less

than `chord_width`, that means the current shading position lies within the chord's path. The shading position is then shaded according to the chords value. If it lies outside the chords path, it will be shaded according to background values. Transition functions, such as `smoothstep()` [15], are then used to interpolate values that lie at the edge of the line stroke. This is done to soften edges and lessen reconstruction artifacts.

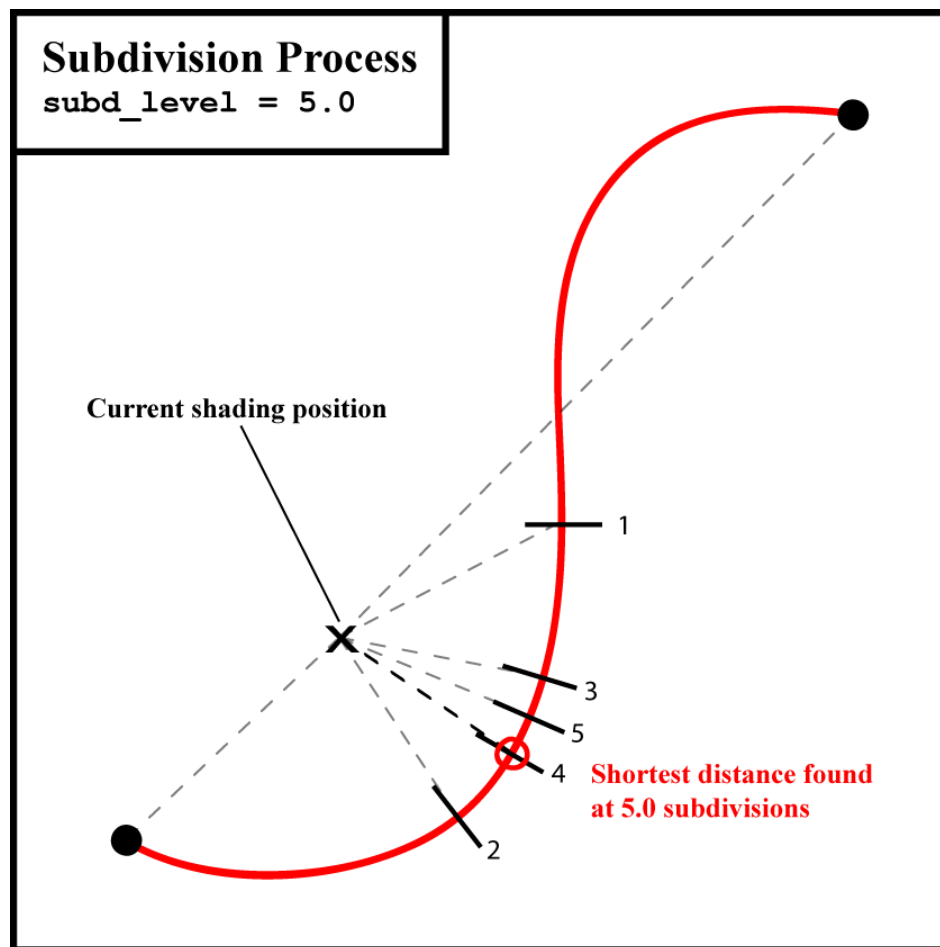


Figure 20: Finding the Shortest Distance between the Spline and the Current Shading Position

The next step of the shading procedure is to evaluate depth information provided by the z component of points along the spline. Information needed for this process is the point on the spline that is closest to the current shading position. Conveniently, this information was already calculated by the previous procedure. The z component of this point should be a value from 0 to 1.0. The closer the value is to 1.0, the more it will protrude, while values near 0 will protrude less. This value will then be calculated along with the value assigned to the current shading position in order to determine that position's depth. Once these final procedures are calculated the result is a single magic number between 0 and 1.0. The magic number should reflect the description of the three-dimensional spline input into the shader when seen visually. This output can now be used to drive shading and displacement processes.

If the shader was output from Maya as a surface shader, the magic number can be used to manipulate color and opacity. O_i and C_i are global variables within the Renderman Shading Language. O_i stores the opacity, and C_i is a color variable containing three components, RGB, to store the final color. If the shader was output as a displacement shader the magic number can be used to modify the position of the point being shaded, P , resulting in a displaced surface. The reason that this magic number is not fed into a specific procedure that manipulates the final color, displacement, or opacity is to maintain the flexibility of the shader. Shading Language users can write their own custom procedures that use the magic number in order to manipulate the color and displacement into a desired shade or form.

CHAPTER IV

RESULTS AND DISCUSSION

The results of this process produce a Renderman shader that can be applied to three-dimensional geometry. This shader can be used to color and displace the rendered appearance of surface meshes in a way that reflects Celtic knots drawn in Maya. The complexity of resulting knot work is dependent on the amount of time a user spends modeling splines. Furthermore, this process is not limited to creating only Celtic knot work; any arbitrary form can be modeled, input into the shader, processed, and reflected in rendered output.

Unexpected results also occurred during the testing of this process. While testing different variations of the `subd_level` parameter, an interesting segmentation effect was discovered. As the `subd_level` parameter is lowered, rendered chords appear broken into segments similar to the appearance of a centipede. When lowered further they break up into dots scattered in uniform intervals along the path of the spline. The segment effect

led to another discovery that a simple offset procedure could be added to the shader producing animatable chord segments. The visual effect of this is that segments or dots appear to move along the path of the spline over time. Procedures were also added that create threads following the path of the spline. The number of threads traveling the chord path is dependent on a thread count defined by the user.

The following images (Figure 21-30) are examples of the shader applied to three-dimensional geometry. Shaders were applied to spheres in order to clearly display the Celtic forms. They were also applied to a short animation produced for this thesis project. The examples exhibit variations of Celtic spline models and shader parameters. Included with the imagery are examples of interesting errors produced while testing the process. These errors were caused by applying shaders to non-uniform meshes including subdivision surfaces. The visual results were wild distortions and shearing of the shader across the surface of geometry.



Figure 21: Celtic Knot Work Shader 01



Figure 22: Celtic Knot Work Shader 02



Figure 23: Celtic Knot Work Shader 03



Figure 24: Celtic Knot Work Shader 04



Figure 25: Shader with Segmentation



Figure 26: Celtic Shader with Threading



Figure 27: Computer Animation without Celtic Shaders



Figure 28: Computer Animation with Celtic Shaders

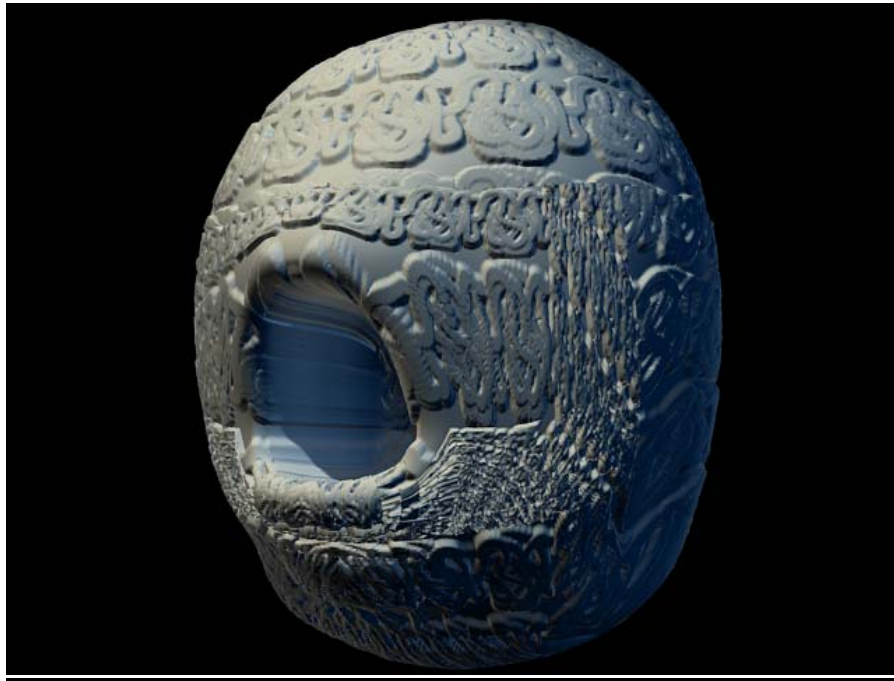


Figure 29: Shading Error 01

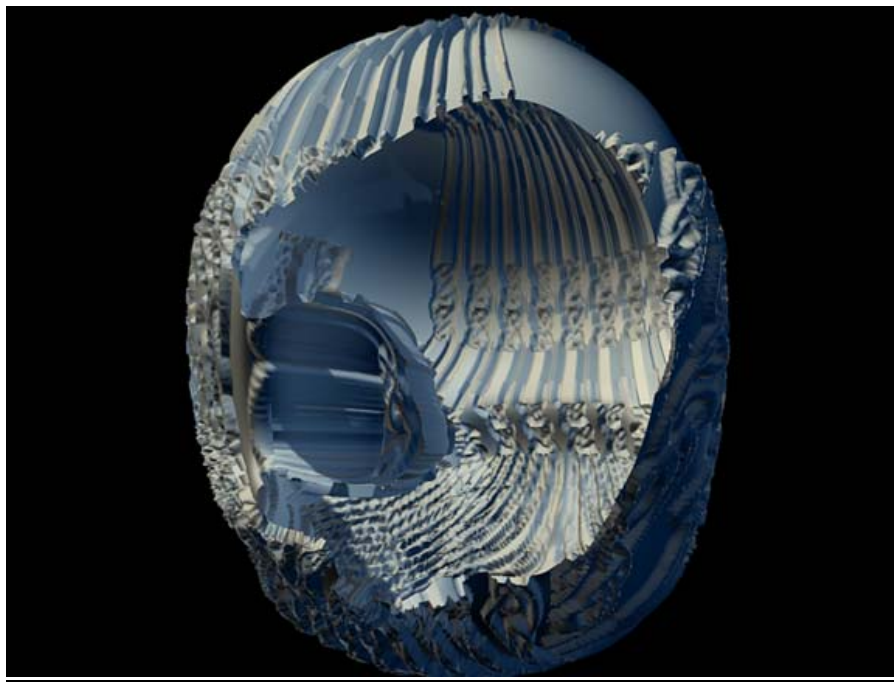


Figure 30: Shading Error 02

Critique of Visual Output

In order to further evaluate the visual result of the knot work process, shaders were attached to a short animation produced by the author of this thesis. The purpose of the animation is to communicate a mystic visual presence and create a stage for Celtic shaders. Content within the animation contains themes inspired by many sources including ancient Celtic Mythology, Celtic art, fantasy art, and the human form. The animation of the scene involves simple deformation, translation, rotation, and scaling of geometry. These actions are used collaboratively in order to create eerie and organic movements. The animation is set within a mysterious landscape contained in a dark vacuum. Within this world are growths of roots that spread across the landscape like searching tentacles. Dwelling underneath the landscape are strange creatures that apparently have only mouths. These creatures push their way through openings in the surface and are fed egg-like food by the roots. The creatures then sink back into the ground and cultivate the earth that the roots feed from. Although the animation only shows a brief interaction between the root and creature, the author wanted to display a symbiotic relationship between them. The author also wanted to explore the idea of reversing environments. The animals of this world dwell within the earth, and the roots of vegetation roam the surface.

Celtic shaders compliment the organic content of the animation by echoing the sweeping contours of root and tongue geometry. Shaders applied to animated geometry also display interesting warping of Celtic forms caused by undulating root movement. The shader deforms with the root thereby stretching and distorting the Celtic designs as

food travels down them. Overall, it is the author's observation that the shaders add unique texture to the surface of geometry and help in achieving an eerie and mystic animation.

Usage of the Celtic Knot Work Process by Other Artists

In order to use the Celtic knot work process, basic knowledge of the Maya and Renderman interface is required. Since the Renderman Shading Language is based on the C programming language, basic programming knowledge would also be helpful. If an artist is familiar with these concepts, he or she should find the process to be intuitive. Artists can use this process within a three-dimensional animation or render out two-dimensional imagery for further manipulation within other computer graphic applications. For example, the graphic appearance of Celtic knot work makes it an interesting element for websites. Knot work can be used as navigational icons or as ornamentation within a web page. Display of Celtic imagery could be viewed within a monitor or atmospherically in the form of a projected installation. Artists could project static or animated imagery into a three-dimensional space, allowing viewers to experience results of the Celtic knot-work process environmentally.

CHAPTER V

CONCLUSIONS

The global purpose of this thesis project is to create visual forms that capture the aesthetic and mystic presence of Celtic Art. The procedural Celtic knot work process was created to aid in the pursuit of this visual presence. Using this process, complex structures can be interactively built, output as procedural shaders, and used to manipulate the rendered appearance of three-dimensional geometry. The strength of this process is that it can sufficiently describe interlaced forms that would otherwise be difficult to create by digitally painting textures. It is also flexible enough to allow for interactive revision, a feature lacked by most procedural generation methods. The aesthetic result is the transformation of smooth surface meshes into layers of interlaced texture. Whether or not this process aids in capturing the mystic presence of Celtic Art is subjective, but the application of these shaders does create unique imagery and changes the visual experience. In addition to Celtic knot work, this process could be used to visualize other complex networks such as the nervous systems of the human body or the tangle of cords and wires behind the workstation of a Visualization student. Animated shaders could also be used to create dynamic Broadway marquees that appear on the façade of theaters or to create completely random forms that simply add visual richness to a three-dimensional environment.

Future Work

Research for further study could involve the development of shaders that use the control points of complete three-dimensional models in order to manipulate the appearance of surfaces. This could be accomplished by creating shading procedures that read in the control points that define an arbitrary surface mesh. Once this information is loaded into the shader, procedures can be developed that interpolate or approximate the control points using spline descriptions. The shader could then color and displace the appearance of geometric surfaces according to the shape of the three-dimensional model. Three-dimensional models embedded into other three-dimensional models could be the visual effect of such a shader.

Animated growth of Celtic knot work from virtual seeds might also yield interesting results. These growths could grow around existing geometry and intertwine with other Celtic knot work growths. Children could branch off parent roots and create children of their own. The visual result could be similar to the growth of ivy around a lattice.

Environmental display of work is another area that could be explored in future research. Celtic imagery could be projected onto dynamic surfaces including the human form, flowing sheets of fabric, and the canopies of trees. The concept of projecting three-dimensional imagery onto three-dimensional surfaces is interesting because a fourth dimension is simulated. A fourth dimension is created by wrapping three-dimensional imagery around three-dimensional surfaces. Projecting imagery onto

volumetric surfaces would not only provide an interesting way of viewing Celtic imagery but would also alter the perception of surrounding environments.

REFERENCES

- [1] R. Allen. *Celtic Art in Pagan and Christian Times*. Dover Publications, Inc., Mineola, NY, 2001.
- [2] A. Apodaca and L. Gritz. *Advanced Renderman: Creating CGI for Motion Pictures*. Morgan Kaufmann Publishers, San Francisco, 1999.
- [3] I. Bain. *Celtic Knotwork*. Constable, London, 1986.
- [4] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Los Altos, CA, 1987.
- [5] C. Davis and D. James. *The Celtic Image*. Blanford Press, London, 1996.
- [6] J. Delyth. Keltic designs. <http://www.kelticdesigns.com>.
- [7] M. Dixon-Kennedy. *Celtic Myth and Legend: An A-Z of People and Places*. Blandford Press, London, 1996.
- [8] B. Froud. The gallery of Brian Froud. <http://beam.to/froud>.
- [9] A. Glassner. *Andrew Glassner's Other Notebook: Further Recreations in Computer Graphics*. A. K. Peters Ltd., Natick, MA, 2002.
- [10] A. Gonzalez. Galerie Sublimatio: Visionary, esoteric art by A. Andrew Gonzalez. <http://www.sublimatrix.com>.
- [11] P. Hanrahan and J. Lawson. A language for shading and lighting calculations. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4): 289 – 298, 1990.

- [12] S. Jones. B - spline parametric curve demonstration.
<http://www.cs.utah.edu/~sjones/spline.html>.
- [13] V. Kruta et al. *The Celts*. Rizzoli International Publications, Inc., New York, 1997.
- [14] L. Laing and J. Laing. *Art of the Celts*. Thames and Hudson Ltd., New York, 1992.
- [15] S. May. RManNotes. <http://www.accad.ohio-state.edu/~smay/RManNotes/rmannotes.html>.
- [16] R. Megaw and V. Megaw. *Celtic Art: From its beginnings to the Book of Kells*. Thames and Hudson Ltd., New York, 1989.
- [17] S. O’Loughlin. The art work of Steve O’Loughlin.
<http://www.stevenoloughlin.com>.
- [18] A. Otstott. Viza616: Rendering and shading. <http://www-viz.tamu.edu/courses/viza616/html/notes.html>.
- [19] D. Salesin, M. Wong, and D. Zongker. Computer-generated floral ornament. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 423 – 434. ACM SIGGRAPH, 1998.
- [20] R. Scharein. The knot plot site.
<http://www.cs.ubc.ca/nest/imager/contributions/scharein/KnotPlot.html>.

VITA

Hunter Guymin Parks
163 Sunset Lake Drive
Huntsville, TX 77340

Education

M. S. in visualization sciences, Texas A&M University,
College Station, Texas, 2003.

B. FA. in computer art, Stephen F. Austin State University, Nacogdoches,
Texas, 2000.